Campus Network Overview

3 Phases of Cisco IIN

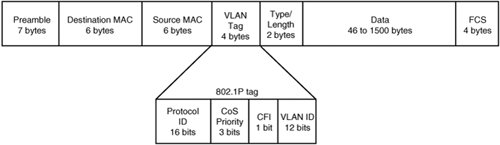
* Integrated transport
* Integrated services
* Integrated applications

3 Layers of SONA

* Networked infrastructure
* Interactive services
* Application

Cisco Enterprise Composite Model

* Enterprise Campus: Building, building dist, core, edge dist, server farm, management
* Enterprise Edge: E-commerce, corporate Internet, VPN and RAS, WAN
* Service Provider Edge: ISP, PSTN, Frame-relay/ATM
* PDNTSPA
* Protocol Data Units
  + 4 (Transport) 🡪 TCP Segment / TCP port
  + 3 (Network) 🡪 Packet / Router
  + 2 (Data Link) 🡪 Frame / Switch/bridge
* Segment 🡪 broadcast domain
* VLAN isolates broadcast domain
* Each switched port is a collision domain



General Routing Concepts

Routing table population process

* Valid next-hop IP 🡪 verify the route has a valid next-hop IP address
* Metric 🡪 Routing protocol chooses the best path based on the lowest metric
* Administrative Distance 🡪 If more than one route exists for same destination and prefix from different routing sources, the lower AD (0 – 255) is selected
* Prefix 🡪 Router looks at prefix advertised; Routes with different prefixes can coexist

RIPv2 Routing Concepts

R1(config)# **router rip**

R1(config-router)# **version 2**

R1(config-router)# **no auto-summary**

R1(config-router)# **network 172.16.0.0**

R1(config-router)# **ip summary-address rip 172.16.0.0 255.255.255.0**

**show ip route rip**

**show ip rip database**

**show ip protocols**

EIGRP Routing Concepts

* Is a hybrid distance-vector protocol (also called advanced distance-vector)
* Benefits
  + Supports backup routes, makes it faster to converge than OSPF and others
  + Allows summarization to happen on any interface, unlike OSPF
  + Guaranteed loop-free topology at all times
  + Only protocol that allows unequal cost load-balancing
* Uses multicast, rather than broadcast, for communications between routers
* EIGRP is transport layer function
* Sits on top of IP; protocol number 88
* Does auto-summarization at classful boundaries by default
* Maintains 3 tables
  + **Neighbor table** 🡪 lists directly connected (adjacent) routers discovered by HELLOs
    - Uses hello packets for neighbor discovery
  + **Topology table** 🡪 Lists best (successor) routes, and next best routes (feasible-successor routes) learned from neighbors
    - Local EIGRP first adds subnets to its topology from 3 sources
      * Prefixes of connected subnets for interfaces on which EIGRP has been enabled on that router using the network command
      * Prefixes of connected subnets for interfaces referenced in an EIGRP neighbor command
      * Prefixes learned by the redistribution of routes into EIGRP from other routing proto- cols or routing information sources
    - When neighbor comes up, updates exchanged
  + **Routing table** 🡪 Contains best route to each destination (current successor)
    - EIGRP offers successor routes only to routing table

**Feasible Distance** 🡪 How far it is from me (router) to a specific route

* Equivalent to sum of AD from next-hop router plus cost between self and next-hop router

**Reported (Advertised) Distance** 🡪 How far my neighbor tells me it is from him to a specific route

**(Current) Successor** 🡪 Primary loop-free path that ends up in routing table

* Chosen because it has the lowest FD of all possible paths to a destination

**Feasible Successor** 🡪 Alternative loop free path to the successor

* Selected at same time as successor, but kept only in topology table
* Topology table can maintain multiple feasible successors for a destination
* Feasibility Condition 🡪 AD < FD of (current) successor

Active Route 🡪 Undergoing re-computation; Router is actively looking for a backup for this route. Bad.

Passive Route 🡪 Not undergoing re-computation; Normal steady-state mode of a route. Good.

5 EIGRP packet types

* HELLO
* UPDATE
* QUERY
* REPLY
* ACKNOWLEDGEMENT

EIGRP Neighborship Process

1. HELLO sent on MCAST 224.0.0.10
2. HELLO sent back on MCAST 224.0.0.10
3. UPDATE sent on MCAST 224.0.0.10
4. UPDATE sent back on MCAST 224.0.0.10

* Metric = [K1 \* BW + ((K2 \* BW) / (256 – load)) + K3 \* delay) \* (K5 / (reliability + K4)]
* K1 = Bandwidth in kbps
* K3 = Delay in microseconds
* K4 and K5 = Reliability
* K2 = Loading
* MTU
* BW = 10^7 / BW
* Real Metric = 256 \* (10^7 / Slowest\_BW + Sum\_of\_All\_Link\_Delays)
* Although EIGRP can use 5 criteria for metric, only 2 are used by default
  + I.e, all K values other than K1 and K3 are set to zero
* If utilized K values don’t match, neighbor won’t form

Implementing EIGRP Routing

router eigrp 90 (start EIGRP process with AS *n*)

network 172.30.0.0 0.0.255.255 (send HELLOs on ints in range and advertise those networks)

no auto-summary (disable auto summarization)

redistribute static

ip default-network [x.x.x.x] (weird way to set default route to a network; must be classful)

passive-interface fa0/0 (disable sending routing updates on interface)

variance 2 (sets unequal cost load-balancing to use routes up to 2x as bad as primary)

* Interface IP must be its primary IP; not allowed to peer with secondary
* Passive interface mode 🡪 Prevents protocol’s routing updates from being sent through the interface
  + No neighbors form on this interface
  + Interface’s subnet may be announced in EIGRP
* EIGRP summarization

int s0/0

ip summary-address eigrp 90 172.30.0.0 255.255.248.0

**show ip eigrp neighbors** [detail] 🡪 shows discovered neighbors

**show ip eigrp topology** 🡪 shows EIGRP topology table; only successor and feasible successor routes are shown

**show ip eigrp topology all-links** 🡪 show all routes including non-successor/FS

**show ip route** [eigrp] 🡪 show EIGRP entries in routing table

* 172.17.0.0/16 [90/40514560] via 192.168.10.1
* Network Admin Distance / EIGRP Metric (FD) next-hop address

**show ip protocols** 🡪 show parameters and state of active routing processes

**show ip eigrp interfaces** 🡪 shows number of peers connected to interfaces

EIGRP Route Filtering

* EIGRP enables route filtering using the distribute-list router subcommand
* The distribute list refers to either an access control list (ACL), prefix list, or route map
* These three tools classify whether a route should be permitted to be sent/received in an EIGRP Update or be denied (filtered)
* The distribute-list command also specifies the direction–outbound updates or inbound updates–and optionally, the specific interface on which to filter updates
* EIGRP route filtering with ACL and distribute-list

**access-list 4 deny 10.17.32.0 0.0.0.255**

**router eigrp 1**

**distribute-list 4 out s0/0**

* EIGRP route filtering with prefix list and distribute-list

**ip prefix-list FRED seq 10 deny 10.17.36.0/24 ge 26 le 26**

**ip prefix-list FRED seq 20 permit 0.0.0.0/0 le 32**

**router eigrp 1**

**distribute-list prefix FRED out**

* EIGRP route filtering with route map and distribute-list
* route-map commands with the permit option either cause a route to be allowed through (if matched by the match command) or remain in the list of routes to be examined by the next route-map clause
* route-map commands with the deny option either filter the route (if matched by the match command) or remain in the list of routes to be examined by the next route- map clause.
* Decision to filter a route or allow the route through is based on the deny or permit in the route-map command, *and not the deny or permit in the ACL or prefix list*
* When referencing an ACL or prefix list from a route map, the ACL or prefix list simply matches all routes permitted by the ACL or prefix list. Routes that are denied by the ACL or prefix list simply do not match that match command’s logic, making IOS then consider the next route-map command
* The route-map command includes an implied deny all clause at the end; to configure a permit all, use the route-map command, with a permit action, but without a match command

**route-map SAMPLE deny 10**

**match ip address prefix-list MYLIST**

**set**

**route-map SAMPLE permit 20**

**match**

**set**

**route-map SAMPLE permit 30**

! Notice – no match commands, so the above clause matches all remaining routes

**router eigrp 1**

**distribute-list route-map out**

EIGRP Route Summarization

**R1(config-int)#** **ip summary-address eigrp** *asn prefix subnet-mask*

**Contiguous network** 🡪 A single classful network in which packets sent between every pair of subnets will pass only through subnets of that same classful network, without having to pass through subnets of any other classful network

**Discontiguous network** 🡪 A single classful network in which packets sent between at least one pair of subnets must pass through subnets of a different classful network.

Auto-summary 🡪 if a router has interfaces in more than one Class A, B, or C network, then that router will advertise a single summary route for an entire Class A, B, or C network into the other classful network, rather than advertise routes for the individual subnets.

Advanced EIGRP Routing

**NBMA** 🡪 non-broadcast multi-access network; typically referring to frame-relay

* Broadcast and multicast traffic are denied on NBMA networks

**DLCI** 🡪 layer 2 data link connection identifier; in frame-relay, analogous to Ethernet MAC

* *DLCI only locally significant*

**PVC** 🡪 permanent virtual circuit, each with its own committed information rate (CIR)

**Split horizon** 🡪 I will not send an advertisement out an interface that I received it on

* Commonly needs to be disabled with frame-relay

int s0/0.1 multipoint

ip address 172.16.124.1 255.255.255.248

frame-relay map ip 172.16.124.2 102 broadcast

frame-relay map ip 172.16.124.3 103 broadcast

show frame-relay map

router eigrp 25

neighbor 172.16.124.2 🡪 manual neighbor; first neighbor statement disables multicast

* Cisco auto disables split horizon if you configure frame-relay on physical interface; split horizon is enabled if you use sub-interfaces

int s0/0.1

no ip split-horizon eigrp 25 🡪 disable split horizon on an interface

* EIGRP limited to 50% of link configured bandwidth by default; this is divided by the number of neighbors if interface is multi-point

int s0/0.1

bandwidth 512

ip bandwidth-percent eigrp 90 60

* EIGRP authentication uses key chains with lifetimes
* NTP sync is important to use to make sure expiries are in sync
* Keys must match string and key number

key chain EIGRP\_KEYS

key 1

key-string cisco1

accept-lifetime 00:00:00 1 Jan 2010 00:00:00 1 Feb 2010

send-lifetime 00:00:00 1 Jan 2010 00:00:00 1 Feb 2010

key 2

key-string cisco2

accept-lifetime 00:00:00 28 Jan 2010 00:00:00 infinite

send-lifetime 00:00:00 28 Jan 2010 infinite

int s0/0.1

ip authentication mode eigrp 25 md5

ip authentication keychain eigrp 25 EIGRP\_KEYS

**debug eigrp packets** 🡪 Useful to troubleshooting key auth problems

EIGRP Design Best Practices

**debug eigrp neighbors**

**debug eigrp packets query reply** 🡪 Will give you only query and reply packets

* Default query response timeout is 3 minutes
* Stuck in active
  + Kills all neighbors and then reestablishes them
* Summary routes
  + Change the way neighbors respond to queries

**Stub Network** 🡪 network with only one way in and out; implication is no reason to ask a stub for any networks other than those it advertises initially

* + Very useful in hub and spoke networks
  + Exempts stub routers from query process
  + Default is advertise connected and summary routes

router eigrp 10

eigrp stub

* Graceful shutdown 🡪 introduced in 12.3(2)2
  + Way to indicate a router is going down without waiting for hold down timer to expire
  + Neighbor sends final HELLO with all K values = 255
  + Essentially a GOODBYE message

OSPF

* Link state protocols
  + Know the entire network topology of their area
  + Send triggered updates when there is a network change
  + Send periodic updates (LS refresh) on long intervals (every 30 minutes)
  + Use Dijkstra’s SPF algorithm
  + Disadvantage is SPF calculation is CPU expensive
  + Only two link-state protocols exist: IS-IS and OSPF
  + Maintain 3 tables
    - Neighbor table 🡪 Tracks all neighbors it has formed a relationship with
    - Topology table 🡪 Complete picture of every network available in my area and all paths to reach them
    - Routing table
* Areas
  + Enables (requires) creation of a hierarchical design
  + Each area contains a specific group of routers, defining a flooding domain
    - All routers in an area have the same topology table
    - Changed inside the area require LSA flooding and full SPF calculation
  + Area 0 is the backbone area
  + All areas *must* connect to area 0; exception is virtual-links
  + Routing between areas hides topology details
    - Inter-area routing is similar to distance vector
    - Changes outside the area don’t always require LSA flooding or SPF
    - Limits impacts on router resources
* OSPF is a two-level hierarchy
  + Backbone area 0 (0.0.0.0)
    - Used to summarize topology information between other areas
    - Traffic from one area to another must pass through 0
    - Must be contiguous
  + Non-backbone areas
    - All other areas 1-2^32 (0.0.0.1 – 255.255.255.255)
    - Must use connections to area 0 to reach other areas

Types of OSPF Routers

**Internal Router** 🡪 router with all interfaces in the same area

**Backbone Router** 🡪 Routers in the perimeter of area 0

**Area Border Router** (ABR) 🡪 router with interfaces in multiple areas

* Will have an interface in area 0 and interface(s) in some other area(s)
* Localizes topology changes
* Used to summarize information between area 0 and non-backbone area

**Autonomous System Boundary Router** (ASBR) 🡪 router with interface(s) in different routing domain

* Used to redistribute information to/from other routing domains and OSPF
* ABR and ASBR are only routers which can do summarization in OSPF

OSPF Packet Types

* HELLO 🡪 Used to discover neighbors and build adjacencies
* Database Description (DBD) 🡪 Used to synchronize the LSDBs
* Link-State Request (LSR) 🡪 Used to request specific link-state records
* Link-State Advertisement (LSA) 🡪 Update about an individual route
* Link-State Update (LSU) 🡪 Contains one or more LSAs; Send LSR’d records
* Link-State Acknowledgement (LSACK) 🡪 Reliability mechanism for OSPF
  + OSPF is its own layer 4 transport protocol 89 (doesn’t use UDP or TCP)
  + Acknowledges all packets except HELLO and LSACK itself

|  |  |  |
| --- | --- | --- |
| LSA Type | Description |  |
| 1 | Router LSA | Each router creates its own Type 1 LSA to represent itself for each area to which it connects. The LSDB for one area contains one Type 1 LSA per router per area, listing the RID and all interface IP addresses on that router that are in that area. Represents stub networks as well. |
| 2 | Network LSA (DR generated) | One per transit network. Created by the DR on the subnet and represents the subnet and the router interfaces connected to the subnet. |
| 3 | Summary LSAs (ABR summary route) | Created by ABRs to represent subnets listed in one area’s type 1 and 2 LSAs when being advertised into another area. Defines the links (subnets) in the origin area, and cost, but no topology data. |
| 4 | Summary LSA (ASBR location) | Like a type 3 LSA, except it advertises a host route used to reach an ASBR. |
| 5 | Autonomous system external LSA (ASBR summary route) | Created by ASBRs for external routes injected into OSPF. |
| 6 | Multicast OSPF LSA | Defined for MOSPF; not supported by Cisco IOS. |
| 7 | Defined for NSSAs | Created by ASBRs inside an NSSA area, instead of a type 5 LSA. |
| 8 | External attributes LSA for BGP | Not implemented in Cisco routers. |
| 9, 10, 11 | Opaque LSAs | Used as generic LSAs to allow for easy future extension of OSPF; for example, type 10 has been adapted for MPLS traffic engineering. |

* LSAs are grouped by 3 route types
  + Intra-Area Routers (O)
    - LSA 1 & 2
  + Inter-Area Routes (O IA)
    - LSA 3 & 4
  + External Routes
    - E1/E2
      * LSA 5
    - N1/N2
      * LSA 7

OSPF Neighbor Process

1. Determine your own router ID
   * Simply the router’s name in the OSPF process
   * Will be highest active IP address when OSPF process starts
     1. Loopbacks beat physical interfaces
   * Also, can (should) be hard coded using router-id command
   * Determined upon rebooting router, or clearing OSPF process
2. Add interfaces to the link state database
   * Dictated by the network command
3. Send a HELLO message on chosen interface(s)
   * Starts in DOWN state
     + Once every 10 seconds on broadcast / P2P networks
     + Once every 30 seconds on NBMA networks
       - Dead timer is 4 x HELLO timer
     + Contains all sorts of info
       - router-id
       - HELLO and dead timers \*
       - Network mask \*
       - Area ID \*
       - Neighbors
       - Router priority
       - DR / BDR IP address
       - Authentication password \*
4. Receive HELLO
   * INIT state
5. Send reply HELLO
   * TWO-WAY state
6. Master slave relationship determined
   * EXSTART state
   * Master simply means who will send its Database Description Packet (DBD) first
     1. Slim preview of all networks
   * Master determined by priority, higher router ID breaks tie
7. DBDs are acknowledged and reviewed
   * LOADING state
   * Slave requests details 🡪 Link State Requests
   * Master sends updates 🡪 Link State Updates
   * Master requests details on networks it is unaware of 🡪 LSRs
   * Slave sends updates 🡪 LSUs
8. Neighbors are synchronized
   * FULL state

* OSPF cost = 100 / Bandwidth (Mbps)
* Default cost metric doesn’t work for > 100Mbit links
  + Platform-dependent defaults
* Be careful not to blow infinite metric
* Should be synced on all routers
* Defaults:
  + 56k = 1785
  + 64k = 1562
  + T1 (1.544) = 65
  + E1 (2.048) = 48
  + Ethernet = 10
  + Fast Ethernet = 1
  + GigE = 1
  + 10GigE = 1
* To adjust for higher-speed links

router ospf 1

auto-cost reference-bandwidth 1000

* BR and BDR elected on broadcast and NBMA networks
* Routers on shared segments will form FULL relationship *only* with DR and BDR
* DR and BDR elected for every shared segment, even if only one router
* DR and BDR not elected on point-to-point links
* DR and BDR listen to multicast 224.0.0.6
* All OSPF routers listen to 224.0.0.5
* DR Election process
  + Highest router priority in HELLO packet; all default to priority 1; 0 prevents
  + Highest router ID
  + This does not preempt if other routers come online later; must clear process or reload
* If the DR fails, the BDR becomes the DR -- regardless of whether a higher priority router has joined the subnet -- and a new election is held to choose a new BDR
* If the BDR fails, a new election is held for BDR, and the DR remains unchanged.
* DROTHER 🡪 Not DR or BDR

Implementing OSPF

* OSPF external route types
  + E1 🡪 increment their metric
  + E2 (default) 🡪 do not increment their metric

router ospf 1

router-id 1.1.1.1

network 172.30.0.1 0.0.0.0 area 0

redistribute static subnets metric 200 metric-type 2

int loopback 1

ip ospf network point-to-point

int fa0/0

ip ospf priority 200 🡪 increase priority to make sure its elected DR

int fa0/0

ip ospf priority 0 🡪 set priority 0 to make sure it’s never elected

show ip ospf neighbor

show ip ospf database

show ip ospf interface

clear ip ospf process

Implementing OSPF over NBMA

3 Network Types

* Broadcast, multi-access networks
  + Ethernet, token ring
  + Single operation mode
  + DR/BDR election, 10s HELLOs, dual multicast address
* Point-to-Point networks
  + T1 CAS, ISDN BRI/PRI
  + Single operation mode
  + No DR/BDR, 10s HELLOs, single multicast address
* Non-broadcast multi-access networks (NBMA)
  + Frame relay, ATM
  + 30s HELLOs
  + These networks deny broadcast and multicast traffic
  + 5 modes of operation

5 OSPF Modes for NBMA

* Non-broadcast mode (NBMA) 🡪 RFC standard
  + - Default mode for X.25, FR, ATM
    - Neighbors statically configured (neighbor 10.21.3.2)
    - Must be one subnet
    - Acts like a LAN environment
    - DR/BDR elected; must have full connectivity to all neighbors
* Point-to-multipoint 🡪 RFC standard
  + - Fixes issues with NBMA networks
    - Requires single subnet
    - No DR/BDR election held
    - Neighbors automatically form, assuming pseudo-broadcasts enabled
* int s0/1.1
  + - ip ospf network point-to-multipoint
    - ip ospf hello-interval 10
* Point-to-point 🡪 Cisco proprietary
  + - Uses separate sub-interfaces
    - Requires different subnets; lots of IPs for lots of neighbors
    - No DR/BDR elected
    - Neighbors automatically form
* Broadcast 🡪 Cisco proprietary
* Point-to-multipoint, non-broadcast 🡪 Cisco proprietary
  + - Just like point-to-multipoint, except neighbors need static assignment

OSPF Area Types and Options

**Virtual Link** 🡪 This is a tunnel to bridge one area to the backbone area, via a transit area

* Perform configuration on both sides of the link
* Temporary fix you can put in place while redesigning networks
* The transit area over which the two routers communicate must not be a stubby area
* **area** argument indicates the area which will serve as transit area

R3.3.3.3(config-router)# **area 1 virtual-link 2.2.2.2**

R2.2.2.2(config-router)# **area 1 virtual-link 3.3.3.3**

R# **sh ip ospf virtual-links**

* For all types of stubby areas, the ABR always filters Type 5 (external) LSAs
* For totally stubby and totally NSSA areas, the ABR also filters Type 3 LSAs
* For stubby and NSSA areas–those without the word “totally” in the name–the ABRs do not filter Type 3 LSAs, advertising Type 3 LSAs as normal

**Stub area** 🡪 Does not accept routes external to the autonomous system

* ABRs create a default route, using Type 3 LSA, listing subnet 0.0.0.0 and mask 0.0.0.0, and flood that into the stub area
* ABRs do not flood (block) Type 5 LSAs into the stub area
* ABRs may not flood other Type 3 LSAs into the area
* All paths external to the autonomous system use the default route

HQ(config-router)# **area 2 stub**

BRANCH(config-router)# **area 2 stub**

**Totally stubby area** 🡪 Does not accept external AS routes, or summary routes from other areas in AS

* All paths external to the area use the default route
* Cisco proprietary (only ABR must be Cisco), other vendors have similar
* Configure **stub** on stub side, **stub no-summary** on ABR side
* Blocks type 3, 4, and 5 LSAs from entering

ABR(config-router)# **area 1 stub no-summary**

BRANCH(config-router)# **area 1 stub**

**Not so stubby area (NSSA)** 🡪 Passes external routes through via type 7 LSAs; these convert back to type 5 LSAs once they reach the backbone

* Similar to stub areas, except NSSAs allow ASBRs, which are prohibited in pure stub areas

HQ(config-router)# **area 1 nssa no-summary**

BRANCH(config-router)# **area 1 nssa**

BRANCH(config-router)# **redistribute rip subnets metric 10**

* Redistribute static routes into OSPF using type E1 (metric does increment)

R1(config-router)# **redistribute static subnets metric-type 1 metric 50**

* Passive interface disables sending routing updates on that interface

R1(config-router)# **passive-interface default**

R1(config-router)# **no passive-interface fa0/0**

* MD5 authentication mode 🡪 allows multiple keys, but all active at once; no lifetime option
* Configure on each interface; both passphrase and key # must match

int fa0/1

ip ospf authentication message-digest

ip ospf message-digest-key 1 md5 aaKue4thoh

* Clear text authentication mode

int fa0/1

ip ospf authentication cisco456

**debug ip ospf adj**

OSPF Filtering

* Filtering can *only* be performed on ABR or ASBR
* OSPF filtering actually involves filtering LSAs
* Since LSDBs in an area must all be the same of OSPF to work, you can’t filter type 1 or 2 LSAs within an area
* Can filter type 3 summary on ABR and type 5 external LSAs on ASBR
* Filtering Type 3 LSAs on ABRs
  + ABRs, by definition, connect to the backbone area and at least one other area
  + ABRs, as a fundamental part of their role as ABR, create and flood Type 3 Summary LSAs into one area to represent the subnets in the other areas connected to that ABR
  + Type 3 LSA filtering tells the ABR to filter the advertisement of these Type 3 LSAs
* Filtering Type 5 LSAs on ASBRs
* Filtering the routes OSPF would normally add to the IP routing table on a single router

OSPF Summarization

* OSPF allows summarization at ABR or ASBR *only*
* Reason is because LSDB must be the same for all routers in an area

Two Types of OSPF Summarization

* Inter-area Route Summarization 🡪 Occurs on ABRs and applies to routes within each area
  + Doesn’t apply to external routes injected via redistribution
  + **area** argument is the origin area from which the subnets exist; summary will be advertised into all other areas connected to the ABR
  + Consolidates type 3 LSAs into a single type 3 LSA
  + Advertises best of metrics by default

**area** *area-id* **range** *ip-address mask* [**cost** *cost*]

ABR(config-router)# **network 172.16.32.1 0.0.0.0 area 1**

ABR(config-router)# **area 1 range 172.16.32.0 255.255.224.0**

* External Route Summarization 🡪 Specific to external routes that are injected into OSPF via route redistribution
  + ASBR is a router that redistributes routes into OSPF from some other routing source
  + ASBR creates a type 5 External LSA for each redistributed subnet
  + ASBR assigns the summary route the metric of the lowest amongst

ASBR(config-router)# **network 172.16.32.1 0.0.0.0 area 1**

ASBR(config-router)# **summary-address 172.16.32.0 255.255.224.0**

OSPF Default Routing

* Two common reasons for injecting default route
  + Drive all Internet destined traffic towards Internet routers
  + Drive traffic inside an area towards one of the ABRs in that area

**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type- value*] [**route-map** map-name]

* Injects a default route into OSPF as an External Type 2 route, using a Type 5 LSA, with metric 1, but only if a default route exists in that router’s routing table

IPv4 Redistribution

**Route redistribution** 🡪 capability of boundary routers connecting different routing domains to exchange and advertise routing information between those routing domains (autonomous systems)

* The main technical reason for needing redistribution is straightforward: An internetwork uses more than one routing protocol, and the routes need to be exchanged between those routing domains, at least temporarily

**Administrative distance** 🡪 used to rate a routing protocol believability (aka, trustworthiness)

* + each routing protocol is prioritized in order from most to least believable using value called administrative distance
  + Lower = better

**Routing metric** 🡪 value representing the path between the local router and the destination network, according to the routing protocol being used

* + used to determine the routing protocol’s “best” path to the destination

|  |  |  |
| --- | --- | --- |
| Route Source | Default Admin Distance | Derived From |
| Connected interface | 0 |  |
| Static route out an interface | 1 |  |
| Static route to a next-hop address | 1 |  |
| EIGRP summary route | 5 |  |
| External BGP | 20 |  |
| EIGRP (internal) | 90 |  |
| IGRP | 100 |  |
| OSPF | 110 | Interface bandwidth |
| IS-IS | 115 | Each interface metric of 10 |
| RIPv1, RIPv2 | 120 | Hop count |
| EGP | 140 |  |
| ODR | 160 |  |
| EIGRP (External) | 170 |  |
| Internal BGP | 200 |  |
| Unreachable | 255 |  |

Seed Metric 🡪 When a router advertises a directly connected network, the initial metric that is used

|  |  |
| --- | --- |
| Distributed Into | Default Seed Metric |
| RIP | 0, interpreted as infinite |
| IGRP/EIGRP | 0, interpreted as infinite |
| OSPF | 20, except BGP routes = 1; all default to type E2 |
| IS-IS | 0 |
| BGP | Set to IGP metric value |

Two Redistribution Techniques

* One-point redistribution 🡪 Only one router redistributes between two routing protocols; Loop safe
  + One-way 🡪 Redistributes only the networks learned from one routing protocol into the other; use a default/static route for other direction
  + Two-way 🡪 Redistributes routes between the two routing processes in both directions
* Multipoint redistribution 🡪 Two separate routers running both routing protocols
  + One-way 🡪
  + Two-way 🡪
* Redistribution is configured under the routing process that is to *receive* the redistributed routes
* **redistribute** command tells the router to take not only routes learned by the source routing protocol, but also connected routes on interfaces enabled with that routing protocol–including passive interfaces
* When the redistribute command refers to another IGP as the routing source, it tells the router to redistribute the following
  + All routes in the routing table learned by that routing protocol
  + All connected routes of interfaces on which that routing protocol is enabled
* Both EIGRP and OSPF treat redistributed routes as external routes
  + OSPF creates a Type 5 LSA to represent each redistributed subnet
  + Except when an NSSA, then creates a Type 7 LSA

Into RIP

**redistribute** *protocol* [*process-id*] [**match** *route-type*] [**metric** *value*] [**route-map** *map-tag*]

Into OSPF

**redistribute** *protocol* [*process-id*] [**metric** *value*] [**metric-type** *value*] [**route-map** *map-tag*] [**subnets**] [**tag** *value*]

Into EIGRP

**redistribute** *protocol* [*process-id* | *as-number*] [**metric** *bandwidth delay reliability loading mtu*] [**match** *route-type*] [**route-map** *map-tag*]

Manipulating Routing Updates

Methods of Manipulating Admin Distance to Influence Route Selection

* Alter default administrative distance with **distance** directive

**distance** *administrative-distance [address wildcard-mask [ip-standard-list] [ip-extended-list]]*

R1(config-router)# **distance 85 192.168.150.0 0.0.0.255**

Route filtering

* Alter default administrative distance with ACL

access-list 1 permit 192.168.150.0 0.0.0.255

router rip

distance 65 0.0.0.0 255.255.255.255 1

* Route tagging
* EIGRP has default infinite metric
* RIP has default infinite metric
* BGP default redistribute metric same as what it was
* OSPF default redistribute metric 20
  + Also default to E2 (do not increment metric)

Distribute lists 🡪 allows an access list to be applied to routing updates

* Drawback, subnet mask cannot easily be matched
* Drawback, ACLs are evaluated sequentially for every IP prefix in the routing update
* Drawback, an extended ACL can be cumbersome to configure

access-list 1 permit 10.1.1.0 0.0.0.255

access-list 1 permit 10.1.3.0 0.0.0.255

access-list 1 permit 10.1.5.0 0.0.0.255

router ospf 1

distribute-list 1 out

**Prefix lists** 🡪 Can be used as a better alternative to ACLs in many route filtering commands

* Perform significantly better than ACLs in loading and route lookup of large lists
* Allow specificity notation e.g. /16 🡪 /24
* One of few commands to allow slash notation

ip prefix list MYLIST permit 10.0.0.0/8 le 24

route-map FILTER\_OSPF\_TO\_EIGRP 10

match ip address prefix-list MYLIST

router eigrp 100

redistribute ospf 1 metric 100 10 10 10 10 route-map FILTER\_OSPF\_TO\_EIGRP

**Route maps** 🡪 Complex access lists that allow some conditions to be tested against the packet or route in question using **match** commands; if the conditions match, some **set** actions can be taken to modify attributes of the packet or route

* Big difference between route maps and ACLs is route maps can modify packet or route
* Applications include route filtering during redistribution, PBR, NAT, BGP

**route-map** *map-tag* [**permit | deny**] [*sequence-number*]

**match ip address** {*access-list-number* | *name*} [*…access-list-number* | *name*] | **prefix-list** *prefix-list-name* [*…prefix-list-name*]

**set ip next-hop** *ip-address* [*…ip-address*]

* Omitting the default parameter gives you logic like this: “Try PBR first, and if PBR’s route does not work, try to route as usual.”
* Including the default parameter gives you logic like this: “Try to route as usual while ignoring any default routes, but if normal routing fails, use PBR.”

Path Control

**Policy based routing** 🡪

* Involves configuring a route map with **match** and **set** commands and then applying the route map to an interface
* Allows more granular routing policies than basic destination-based routing using the routing table

ip access-list extended CLIENT1

permit ip host 192.168.1.20 any

!

route-map POLICY permit 10

match ip address CLIENT1

set ip next-hop 201.1.1.2

!

! match telnet and HTTPS

ip access-list extended CLIENT2

permit tcp host 192.168.1.21 any eq 23

permit tcp host 192.168.1.21 any eq 443

!

route-map POLICY permit 20

match ip address CLIENT2

set ip next-hop 200.1.1.2

!

! no match clause means match anything

route-map POLICY permit 30

set ip next-hop 201.1.1.2

!

int f0/0

ip policy route-map POLICY

!

show route-map

IP SLA 🡪

* Create a monitor operation and assign it an integer operation number
* Define the operation type and parameters
* Define frequency
* Define schedule
* Track object

ip sla 100

icmp-echo 200.1.1.2 source-interface fa0/0

timeout 1000

frequency 5

exit

!

ip sla schedule 100 start-time now life forever

track 1 rtr 100 reachability

!

ip access-list extended ROUTER

permit ip any any

!

route-map ROUTER-TRAFFIC permit 10

match ip address ROUTER

set ip next-hop verify-availability 200.1.1.2 10 track 1

set ip next-hop 201.1.1.2

!

ip local policy route-map ROUTER-TRAFFIC

!

show ip sla configuration

!

show ip sla statistics

BGP

* Only external gateway protocol commonly in use
* BGP runs on top of TCP port 179; uses TCP ACK for its reliability mechanism
* Metric is compound of attributes
* BGP technically a distance-vector protocol, but usually known as a path vector protocol
  + By default, operates like RIP hop-count (AS path count)
* Does not require hierarchical topology
* BGP is slow to converge by design because of scale of tables

Homing Types

* Single homed 🡪 Single link
* Dual homed 🡪 Dual links to same ISP
* Multi-homed 🡪 Single links to two different ISPs
* Dual multi-homed 🡪 Dual links to two different ISPs

4 BGP message / packet types

* OPEN 🡪 first message sent after TCP handshake between neighbors
* KEEPALIVE 🡪 sent every 60 seconds by default – resets hold down timer
* UPDATE 🡪 an update for a single path
* NOTIFICATION 🡪 sent when BGP detects and error condition – these close cause a session to close

BGP Tables

* Neighbor table 🡪 Contains all *configured* BGP neighbors – not only connected
* BGP table 🡪 a.k.a., BGP topology table, BGP topology database, BGP routing table, BGP forwarding database
  + Separate from routing table
  + Offers best routes to routing table
* Routing table
* IBGP 🡪 BGP running between routers in the same autonomous systems
* EBGP 🡪 BGP running between routers in different autonomous systems
* Use update-source loopback 0 with loopbacks

BGP States

* Idle 🡪 Searching for neighbors
* Connect 🡪 TCP three-way handshake complete with neighbor
* Open Sent 🡪 BGP open message has been sent
* Open Confirm 🡪 Response received
* Established 🡪 BGP neighborship is established

**BGP speaker** 🡪 Any router that runs BGP is known as a BGP speaker

**BGP neighbor (peer)** 🡪 a BGP speaker that is configured to form a neighbor relationship with another BGP speaker for the purpose of directly exchanging BGP routing information

* BGP neighbors (peers) configured statically
* BGP neighbors are *not necessarily* directly connected
* EBGP neighbors are *usually* directly connected
* Updates are triggered and incremental; no periodic updates

**Autonomous system** 🡪 a set of routers under a single technical administration, using an IGP and common metrics to determine how to route packets within the AS, using an inter-AS routing protocol to determine how to route packets to other AS

* Public AS range: 1 - 64511
* Private AS range: 64512 - 65535

**Transit autonomous system** 🡪 an autonomous system that routes traffic from one external AS to another external AS

BGP route selection process

**0. Ignore routes with an inaccessible next hop address**

1. **Prefer the path with the highest WEIGHT (default 0)**
2. **Prefer the path with the highest LOCAL\_PREF (default 100)**
3. **Prefer the path that was locally originated via a network command**
4. **Prefer the path with the shortest AS\_PATH**
5. **Prefer the path with the lowest origin type**
6. **Prefer the path with the lowest multi-exit discriminator (MED)**
7. Prefer EBGP over IBGP
8. Prefer the path with the lowest IGP metric to the BGP next hop
9. Determine if multiple paths require installation in the routing table for BGP Multipath
10. When both paths are external, prefer the path that was received first (the oldest one)
11. Prefer the route that comes from the BGP router with the lowest router ID
12. If the originator or router ID is the same for multiple paths, prefer the path with the minimum cluster list length
13. Prefer the path that comes from the lowest neighbor address

Three Common Multi-homing Options

* Each ISP passes only default route
* Each ISP passes only default route and provider owned space
* Each ISP passes full tables

N = next hop reachability  
W = weight, bigger is better  
L = local preference, bigger is better  
L = locally injected preferred over BGP learned  
A = AS path length, shorter is better  
O = origin, (igp is better than egp is better than incomplete)  
M = MED, lower is better  
N = neighbor type, ebgp better than ibgp  
I = IGP metric to BGP next-hop, lower is better

Implementing and Tuning BGP

* Routers can only belong to one BGP AS
* BGP router will not accept routing updates that already contain its AS number

**BGP peer group** 🡪 a group of BGP neighbors that all have the same update policies

* Recommended when a BGP router has many neighbors

**neighbor** *peer-group-name* **peer-group**

**router bgp** *autonomous-system-id*

**neighbor *{****neighbor-ip | peer-group-name}* **remote-as** *remote-as-id*

* For IBPG, only difference is both sides use same AS
* Use loopback interfaces for neighbor relationships in IBGP
* No point in using loopbacks for EBGP because they are usually point-to-point links

int loopback 1

ip address 1.1.1.1 255.255.255.255

router ospf 1

network 1.1.1.1 0.0.0.0 area 0

router bgp 5500

neighbor 4.4.4.4 remote-as 5500

neighbor 4.4.4.4 update-source loopback 1

int loopback 4

ip address 4.4.4.4 255.255.255.255

router ospf 1

network 4.4.4.4 0.0.0.0 area 0

router bgp 5500

neighbor 1.1.1.1 remote-as 5500

neighbor 1.1.1.1 update-source loopback 4

Distribute List

access-list 5 permit 192.168.0.0 0.0.1.255

router bgp 100

neighbor 1.1.1.1 distribute-list 5 out

Inject Default Route via BGP

router bgp 100

neighbor 1.1.1.1 default-originate

Verification Commands

show ip bgp

show ip bgp summary

show ip bgp rib-failure

show ip bgp neighbors

debug ip bgp {dampening | events | keepalives | updates}

BGP Attributes

* Attributes are tags that you can assign to incoming or outgoing BGP routes
* Some attributes are WELL-KNOWN (universally supported), some are OPTIONAL
  + Mandatory
    - AS path
    - Next hop address
    - Origin
* Some attributes are MANDATORY (must be in the update), other are DISCRETIONARY
* Some attributes are TRANSITIVE (travel from AS to AS), other are NON-TRANSITIVE

Branch / Mobile Routing

IPSEC Site-to-site VPN Configuration Steps

1. Configure initial key (ISAKMP) details
2. Configure IPSEC details
3. Configure crypto ACL
4. Configure VPN tunnel details
5. Apply crypto map to an interface

**show crypto map** {**interface** *interface-name*} {**tag** *map-name}*

**show crypto session** {**detail**} *ip-address*

**show crypto ipsec sa**

**debug crypto ipsec**

**GRE (generic routing encapsulation)** 🡪 tunneling protocol developed by Cisco for encapsulating network layer protocols packets inside IP tunnels

* Common for passing dynamic routing protocols over IPSEC

IPv6 Concepts

* RFC 2460
* 12.2(2)T and later support
* Address length moved from 32-bit (IPv4) to 128-bit (IPv6)
* Provides 3.4 x 10^38 addresses
* One big goal is to eliminate NAT entirely
* To make addresses more manageable, divided into 8 groups of 4 hex characters each

2001 : 0050 : 0000 : 0000 : 0000 : 0AB4 : 1E2B : 98AA

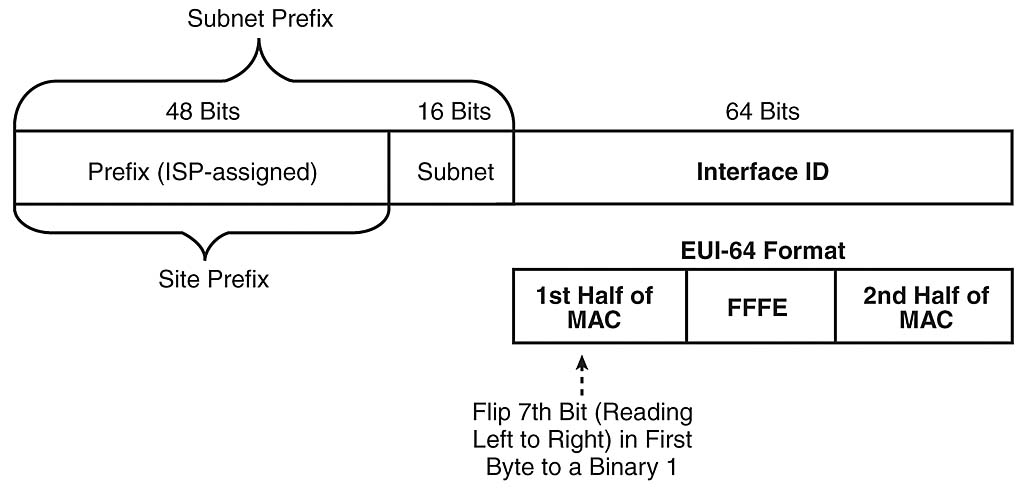
* Rule 1: Eliminate groups of consecutive zeros, ONCE per address

2001 : 0050 :: 0AB4 : 1E2B : 98AA

* Rule 2: Drop leading zeros

2001 : 50 :: AB4 : 1E2B : 98AA

* Loopback address is ::1
* IPv6 header is quite simplified
* Header extensions are supported
* Unicast 🡪 one-to-one
* Multicast 🡪 one-to-many
* Anycast 🡪 one-to-closest
* No broadcast traffic in IPv6; replaced by multicast
* Link-local scope address 🡪 layer 2 domain
  + Assigned automatically as an IPv6 host comes online
  + Similar to 169.254.x.x addresses in IPv4
  + Always begin with “FE80” (first 10 bits: 1111 1110 10) followed by 54 bits of zeros
  + Last 64 bits is the 48-bit MAC address with “FFFE” squeezed in the middle
* Unique / site-local scope address 🡪 organization (equiv. of RFC 1918 addresses); *deprecated*
* Global scope address 🡪 Internet
  + Have their high-order 3 bits set to 001 (2000::/3)
  + Global routing prefix is 48 bits or less
  + Subnet ID is comprised of whatever bits are left over after global routing prefix
  + The primary addresses expected to comprise the IPv6 Internet are from the 2001::/16 subnet



* Stateless Auto-configuration
* EUI 🡪 Extended unique identifier

Implementing IPv6

int lo 50

ipv6 address 2001:5511:5522:AAAA:BBBB:!!!!:2345:234/64

show ipv6 interface lo 50